бюджетное профессиональное образовательное учреждение Вологодской области

«Череповецкий лесомеханический техникум им. В.П. Чкалова»

Специальность **09.02.07** «Информационные системы и программирование»

**ОТЧЕТ ПО ПРОИЗВОДСТВЕННОЙ ПРАКТИКЕ**

**ПП по ПМ.02 Осуществление интеграции программных модулей**

Выполнил студент 2 курса группы ИС-\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

подпись \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

место практики \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

наименование юридического лица, ФИО ИП

Период прохождения:

с «\_\_\_» \_\_\_\_\_\_\_ 2025 г.

по «\_\_\_» \_\_\_\_\_\_\_ 2025 г.

Руководитель практики от

предприятия

должность\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

подпись\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

МПРуководитель практики от

техникума: Материкова А.А.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Оценка:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

«\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2025 года

г. Череповец

2025

СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 3](#_Toc212648300)

[1 ОБЩАЯ ХАРАКТЕРИСТИКА ПРЕДПРИЯТИЯ 4](#_Toc212648301)

[1.1 Организационная структура предприятия 4](#_Toc212648302)

[1.2 Внутренний распорядок работы предприятия, охрана труда на предприятии (организации) 5](#_Toc212648303)

[1.3 Должностные инструкции ИТ-специалистов предприятия 6](#_Toc212648304)

[2 ОСУЩЕСТВЛЕНИЕ ИНТЕГРАЦИИ ПРОГРАММНЫХ МОДУЛЕЙ 8](#_Toc212648305)

[2.1 Разработка требования к программным модулям на основе анализа проектной и технической документации на предмет взаимодействия компонент 8](#_Toc212648306)

[2.2 Выполнение интеграции модулей в программное обеспечение 9](#_Toc212648307)

[2.3 Выполнение отладки программного модуля с использованием специализированных программных средств 10](#_Toc212648308)

[2.4 Осуществление разработки тестовых наборов и тестовых сценариев для программного обеспечения 10](#_Toc212648309)

[2.5 Инспектирование компонентов программного обеспечения на предмет соответствия стандартам кодирования 11](#_Toc212648310)

[3 ВЫПОЛНЯЕМЫЕ ЗАДАНИЯ 12](#_Toc212648311)

[3.1 Разработка программного модуля обработки изображений 12](#_Toc212648312)

[3.2 Интеграция программных компонентов и тестирование 13](#_Toc212648313)

[3.3 Результаты работы и перспективы развития проекта 14](#_Toc212648314)

[3.4 Инспектирование кода и улучшение 15](#_Toc212648315)

[3.5 Формирование проектной документации 15](#_Toc212648316)

[ЗАКЛЮЧЕНИЕ 16](#_Toc212648317)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 17](#_Toc212648318)

[ПРИЛОЖЕНИЯ 18](#_Toc212648319)

# 

# ВВЕДЕНИЕ

Производственная практика проходила в ООО Малленом Системс

Срок практики: с 20.10.2025 по 02.11.2025

Место прохождения практики: ООО Малленом Системс

Студент: Кузнецов Илья Александрович, группа ИС-21

Руководитель практики: Южакова Надежда Витальевна

Цели и задачи во время прохождения производственной практики:

1. Разрабатывать требования к программным модулям на основе проектной и технической документации на предмет взаимодействия компонент
2. Выполнять интеграцию модулей в ПО
3. Выполнять отладку программного модуля с использованием специализированных программных средств
4. Осуществлять разработку тестовых наборов и тестовых сценариев для ПО
5. Производить инспектирование компонент ПО на предмет соответствия стандартам кодирования

# 

# 1 ОБЩАЯ ХАРАКТЕРИСТИКА ПРЕДПРИЯТИЯ

«Малленом Системс» - ведущая российская компания в области разработки и внедрения систем компьютерного зрения, промышленной видеоаналитики и интеллектуальной обработки данных, основанная в 2011 году.

Компания поставляет как готовые программные продукты и программно-аппаратные комплексы (ПАК), так и ведет заказную разработку систем видеоаналитики под требования заказчика.

Основные цели компании – это открытие новых возможностей для бизнеса, которые создают системы видеоаналитики и интеллектуальной обработки данных, тем самым повышая безопасность и эффективность технологических процессов в широком спектре отраслей.

Благодаря индивидуальному подходу, ответственности и глубокому уровню экспертизы, идет создание надежных и качественных решений, которые обеспечивают достижение бизнес-целей клиентов.

# Организационная структура предприятия

Компания «Малленом Системс» представляет собой линейно-функциональную структуру управления, которая сочетает в себе принципы специализации управленческих функций.

Организационная структура: Линейная, есть отделы компании и в них есть руководители. Краткая характеристика каждого отдела:

Центр по развитию интеллектуальных систем, отдел разработки ПО: Проектирование, разработка, оптимизация ПО для клиентов компании.

Производственно-технический отдел: Отдел с инженерами, которые производят пусконаладочные работы на предприятиях.

Административно-управленческий персонал: Руководство компании, которое формирует стратегии развития, управляет отделами, планирует деятельность предприятия, обеспечивает внешние коммуникации компании на выставках.

Группа Маркетинга: Формирование маркетинговой стратегии компании, продвижение бренда и продуктов на рынке.

Коммерческий отдел: Продажа продуктов компании заказчикам, поиск новых клиентов.

Отдел технической поддержки и контроля качества: Техническая поддержка пользователей и тестировка ПО на выявление ошибок и проблем.

Отдел акселерационных и образовательных программ: Разработка и проведение обучающих курсов по машинному зрению и языку программирования.

Отдел кадров: Управление персоналом компании, поиск, подбор, адаптация сотрудников, ведение кадрового документооборота.

Юридический отдел: Обработка всех документов в компании в соответствии с законодательством, взаимодействие с заказчиками и менеджерами по договору.

Бухгалтерия: Ведение экономической деятельности предприятия, бухгалтерского учета, формирование бюджетов компании.

Общественно-хозяйственные рабочие: Поддержание чистоты, порядка на рабочих местах, ремонт, уборка служебных помещений.

# Внутренний распорядок работы предприятия, охрана труда на предприятии (организации)

В компании «Малленом Системс» действует график работы 5/2, с 09:00 до 18:00. В штате компании есть отдельный специалист по охране труда, который проводит вводные инструктажи при приеме на работу и практике, а также занимается выдачей пропусков для пусконаладочных работ инженеров.  
Так же производится обучение сотрудников по промышленной безопасности, по электробезопасности, охране труда и обучению в области применения средств защиты и оказания первой медицинской помощи в «Учебный центр Экоконсалт».

Производится установка оборудования на площадках предприятий по всей России от металлургической промышленности, нефте-газохимической, до производства детского питания.

# Должностные инструкции ИТ-специалистов предприятия

В компании разработаны должностные инструкции для всех должностей. На примере должностей инструкции техника рассмотрим обязанности ИТ-специалистов.

Общие положения:

* Техник относится к категории специалистов
* Техник принимается на работу и увольняется приказом Гендиректора
* Техник подчиняется непосредственно руководителю структурного подразделения, ведущему программисту или руководителю проекта
* На должность назначается лицо без предъявления требований к образованию и опыту работы

Основные требования к знаниям и умениям:

* Основы программирования и тестирования
* Анализ и документирование
* Работа с кодом
* Внутренние правила

Должностные обязанности: Проведение технических расчетов, наладка и проверка оборудования; участие в экспериментах, испытаниях и разработке технической документации; сбор, обработка и систематизация исходных материалов и данных; оформление плановой и отчетной документации, внесение изменений в технические документы; использование в работе современных технических средств.

Права: запрашивать информацию и разъяснения, необходимые для выполнения работы; участвовать в обсуждении проектов и вопросов, связанных со своими обязанностями; вносить предложения по улучшению организации труда; знакомиться с документами, определяющими его права и обязанности.

Несет ответственность за: ненадлежащее исполнение своих должностных обязанностей; разглашение коммерческой тайны; нарушение правил внутреннего трудового распорядка, охраны труда и пожарной безопасности; причинение ущерба организации в установленном законом порядке.

# 2 ОСУЩЕСТВЛЕНИЕ ИНТЕГРАЦИИ ПРОГРАММНЫХ МОДУЛЕЙ

# 2.1 Разработка требования к программным модулям на основе анализа проектной и технической документации на предмет взаимодействия компонент

Описание программы и модулей

Назначение программы: Программа для преобразования цветных изображений в черно-белые с возможностью сохранения результатов в указанную пользователем директорию.

Архитектура программы состоит из 5 основных модулей:

1. Модуль ввода данных (Data Input Module)

* Назначение: Загрузка и валидация исходных изображений
* Функциональность: Чтение файлов формата JPG, PNG, BMP; проверка существования и доступности исходного файла; обработка ошибок файловой системы

1. Модуль преобразования изображений (Image Processing Module)

* Назначение: Конвертация цветного изображения в градации серого
* Функциональность: Преобразование цветового пространства RGB – Grayscale (оттенок серого); сохранение исходного разрешения и метаданных; обработка исключений поврежденных файлов

1. Модуль управления путями (Path Management Module)

* Назначение: Работа с файловой структурой и директориями
* Функциональность: Автоматическое создание целевых директорий; проверка прав доступа на запись; обработка спецсимволов в папках

1. Модуль сохранения результатов (Output Module)

* Назначение: Сохранение обработанных изображений
* Функциональность: Запись файлов в указанную директорию; сохранение в формате, аналогичном исходному; контроль целостности данных

1. Модуль логирования (Logging Module)

* Назначение: Информирование пользователя о процессе работы
* Функциональность: Вывод статусов выполнения операций; форматирование сообщений об ошибках; отображение путей к результатам

Взаимодействие модулей:

Программа реализует последовательную pipeline-архитектуру (друг за другом)

Ввод-преобразование-управление путями-сохранение-логирование.

# 2.2 Выполнение интеграции модулей в программное обеспечение

Интеграция модулей системы обработки изображений проводилась поэтапно с использованием принципа постепенного наращивания функциональности. На начальном этапе был построен базовый контур, включающий модули чтения исходных данных и их предварительной обработки. Особое внимание уделялось обеспечению совместимости между компонентами, на основе библиотеки Pillow, и стандартными модулями Python для работы с файловой системой.

После успешного тестирования к системе подключались модули преобразования цветовых пространств и управления путями сохранения данных. Для стабильного взаимодействия компонентов была сделана система обработки исключений и промежуточных буферов данных. Каждый новый модуль проходил интеграционное тестирование совместно с работающими компонентами.

На последнем этапе проводилось полное тестирование интегрированной системы, в ходе которого проверялось выполнение операций и устойчивость к неправильным входным данным.

# 2.3 Выполнение отладки программного модуля с использованием специализированных программных средств

Процесс отладки программного модуля обработки изображений проводился с использованием интегрированной средой разработки Visual Studio Code и встроенного отладчика Python. Основной упор уделялся диагностике ошибок обработки графических данных и исключений файловой системы.

С помощью модуля logging была реализована система детального логирования, фиксирующая все этапы работы программы – от открытия исходного файла до сохранения результата. Это позволило выявить и исправить ошибки, связанные с неправильной обработкой путей к файлам и исключениями доступа.

Для автоматизации тестирования использовалась команда import os, где были созданы тестовые случаи для проверки обработки изображений разных форматов, работы с неправильными путями и обработки исключительных ситуаций. В результате отладки была достигнута стабильная работа модуля, обеспечена правильная обработка ошибок и оптимизировано потребление системных ресурсов.

# 2.4 Осуществление разработки тестовых наборов и тестовых сценариев для программного обеспечения

Разработка тестовых наборов и сценариев для программного модуля обработки изображений проводилась с использованием модульного тестирования на основе библиотеки Pillow. Были созданы тестовые случаи, включающие тестирование модуля чтения изображений, модуля преобразования цветовых пространств и модуля работы с файловой системой.

Для проверки правильной обработки изображений различных форматов были подготовлены тестовые наборы, включающие файлы JPG, PNG и BMP с различными разрешениями и цветовыми профилями. Тестовые сценарии включали проверку граничных условий, таких как обработка изображений с максимально допустимыми размерами, работа с поврежденными файлами.

Также были разработаны интеграционные тестовые сценарии, проверяющие корректность передачи данных между компонентами системы. В процессе тестирования использовались как автоматизированные тесты, так и ручная проверка визуального качества обработанных изображений.

# 2.5 Инспектирование компонентов программного обеспечения на предмет соответствия стандартам кодирования

В процессе инспектирования компонентов программного обеспечения проводилась проверка соответствия кода общепринятым стандартам написания программ на Python. Основное внимание уделялось соблюдению правил оформления кода, включая использование правильных отступов, соблюдение максимальной длины строки, расстановка пробелов вокруг операторов и после запятых.

Была проведена проверка именования переменных и функций на соответствие требованиям Python Enhancement Proposal 8. Для переменных и функций используется стиль snake\_case, названия отражают их назначение и легко читаются. Также проверил наличие и качество комментариев к коду, которые должны пояснять сложные алгоритмы и назначение функций.

Особое внимание было к организации импортов в начале файла и логической структуре программы. Проверилось, что код разделен на логические блоки.

3 ВЫПОЛНЯЕМЫЕ ЗАДАНИЯ

Прохождение производственной практики началось с создания репозитория на GitHub с четкой структурой для хранения файлов и организации рабочего процесса. Это позволило с первого дня соблюдать порядок в документации и исходном коде, что особенно важно при работе над проектами в команде.

Организация рабочего пространства включала создание отдельных папок для исходного кода, тестовой документации, примеров изображений и отчетной документации, что в дальнейшем значительно упростило навигацию по проекту и позволило легко ориентироваться в структуре файлов даже спустя несколько недель активной разработки.

# 3.1 Разработка программного модуля обработки изображений

На первой неделе практики была поставлена задача разработки программного модуля для обработки изображений. После полного анализа требований стало ясно, что необходимо реализовать инструмент, который может преобразовывать цветные изображения в черно-белые, сохранять их в указанную директорию и формировать резервные копии исходных файлов. Особое внимание уделялось тому, чтобы программа могла работать с различными графическими форматами, включая JPEG, PNG и BMP, при этом сохраняя качество изображения после конвертации.

В качестве основного языка разработки был выбран Python благодаря простоте и наличию мощных библиотек для работы с изображениями. Также использовалась интегрированная среда разработки Visual Studio Code, что упростила ход работы. Библиотека Pillow предоставила все необходимые инструменты для загрузки, обработки и сохранения графических файлов.

В процессе разработки была реализована модульная архитектура, включающая пять основных компонентов:

* Модуль ввода данных (Data Input Module) – отвечает за получение и обработку исходных параметров. Для работы с ОС использован модуль import os, обеспечивающий кроссплатформенную совместимость.
* Модуль преобразования изображений (Image Processing Module) – ядро системы, выполняющее конвертацию цветного изображения в черно-белое. Основой модуля стала библиотека from PIL import image, предоставляющая мощные инструменты для работы с графическими файлами.
* Модуль управления путями (Path Management Module) – использует библиотеку os.path для корректной работы обработки и формирования путей к файлам в разных ОС.
* Модуль сохранения (Output Module) – отвечает за физическое сохранение результатов работы с использованием методов библиотеки PIL.
* Модуль логирования (Logging Module) – предоставляет пользователю детальную информацию о ходе выполнения операций через консольные сообщения, включая пути к созданным файлам, статус выполнения операций и возможные ошибки обработки.

Разработанный модуль process\_image() стал центральным элементом всей системы, контролируя процесс от открытия исходного файла до сохранения финальных результатов. Например, при обработке файла «photo.jpg» программа создает два новых файла: «photo\_original.jpg» (копия исходника) и «photo\_black\_white.jpg» (обработанная версия)

# 3.2 Интеграция программных компонентов и тестирование

На второй неделе упор сместился на тестирование, интеграцию и отладку модулей. Все началось с инспектирования кода, проверяя соответствие стандартам PEP8, оценивая эффективность алгоритмов и выявляя потенциальные уязвимые места. В ходе Code Review обнаружилось, что код хорошо написан, но требует дополнительной обработки исключений для случаев, когда исходный файл не существует или поврежден.

Процесс интеграции компонентов потребовал тщательной настройки взаимодействия между модулем обработки изображений, файловой системой и пользовательским интерфейсом. Была проведена серия тестов, проверяя как программа обрабатывает различные сценарии – от работы со стандартными изображениями до обработки файлов с нестандартными расширениями или недостаточными правами доступа.

Тестирование функциональности включало проверку всех аспектов работы программы. Были разработаны и успешно выполнены тестовые сценарии для модуля обработки изображений (проверка корректности преобразования в черно-белое, создание резервных копий и формирования имен выходных файлов) и модуля взаимодействия с пользователем (проверка обработки путей к файлам, вывода информационных сообщений). Например, при тестировании конвертации в черно-белое использовались изображения разных форматов и размеров, убеждаясь, что программа сохраняет пропорции и качество. Тестирование работы с файловой системой проводилось на примерах путей разной глубины вложенности и с использованием специальных символов в именах файлов.

# 3.3 Результаты работы и перспективы развития проекта

В результате за 2 недели практики был разработан, протестирован и полностью готов к использованию модуль обработки изображений. Программа успешно обрабатывает различные типы графических файлов, корректно работает с файловой системой и предоставляет пользователю понятные сообщения о ходе выполнения операций. Модульная архитектура позволяет легко расширять функциональность – например, добавлять новые фильтры или поддерживать дополнительные форматы файлов.

Перспективы развития проекта включают возможность добавления пакетной обработки изображений, когда пользователь может выбрать целую папку с файлами для конвертации, реализацию дополнительных фильтров (инверсия цветов, повышение резкости), а также создание графического интерфейса для более удобного взаимодействия с программой.

# 3.4 Инспектирование кода и улучшение

На завершающем этапе разработки было проведено полное инспектирование исходного кода с проверкой соответствия стандартам PEP8. Анализ включал оценку именования переменных и функций, проверку отступов и структуры кода, а также оптимизацию рабочих алгоритмов. В процессе улучшения была оптимизирована архитектура программы: дублирующиеся операции вынесены в отдельные функции, добавлены комментарии к ключевым блокам кода. Особое внимание уделялось усилению валидации входных данных и улучшению обработки исключительных ситуаций.

# 3.5 Формирование проектной документации

В ходе практики была разработана полная комплектация проектной документации, включающая техническое задание, описание архитектуры системы, тестовую документацию и пользовательские инструкции. Документация оформлена в соответствии с требованиями предприятия и включает в себя подробное описание всех модулей системы, API интерфейсов и протоколов взаимодействия между компонентами.

ЗАКЛЮЧЕНИЕ

В ходе производственной практики были успешно выполнены все поставленные задачи и достигнуты цели, связанные с разработкой программного модуля для обработки изображений.

Практика позволила не только закрепить теоретические знания, но и приобрести ценный практический опыт в области программной инженерии.

Основные результаты работы включают:

* Разработан полнофункциональный модуль обработки изображений на языке Python с использованием библиотеки Pillow
* Реализована модульная структура системы, включающая пять взаимосвязанных компонентов
* Проведено комплексное тестирование ПО с составлением тестовых сценариев
* Выполнена интеграция всех модулей в единую систему
* Проведено инспектирование кода и его оптимизация в соответствии со стандартами PEP8
* Сформирована полная комплектация проектной документации

В процессе работы были освоены и применены на практике профессиональные компетенции ПК 2.1-2.5, включая разработку требований к программным модулям, интеграцию компонентов, отладку ПО, разработку тестовых сценариев и инспектирование кода.

Полученный опыт подтвердил важность модульного подхода в разработке программного обеспечения, который обеспечивает гибкость, масштабируемость и удобство сопровождения кода.

Реализованная система обладает значительным потенциалом для дальнейшего развития – возможно добавление пакетной обработки изображений, новых фильтров и эффектов, а также создание графического интерфейса пользователя.

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Python Software Foundation. Официальная документация Python [Электронный ресурс] – режим доступа: [https://docs.python.org/3/](https://docs.python.org/3/%20)
2. Pillow (PIL Fork) Documentation [Электронный ресурс] – режим доступа: [https://pillow.readthedocs.io/en/stable/](https://pillow.readthedocs.io/en/stable/%20)
3. PEP 8 – Style Guide for Python Code [Электронный ресурс] – режим доступа: [https://www.python.org/dev/peps/pep-0008/](https://www.python.org/dev/peps/pep-0008/%20)
4. Software Testing Fundamentals [Электронный ресурс] – режим доступа: [https://www.guru99.com/software-testing.html](https://www.guru99.com/software-testing.html%20)
5. Интеграционное тестирование ПО [Электронный ресурс] – режим доступа: [https://habr.com/ru/articles/282582/](https://habr.com/ru/articles/282582/%20)
6. Отладка программного обеспечения [Электронный ресурс] – режим доступа: [https://habr.com/ru/articles/205426/](https://habr.com/ru/articles/205426/%20)
7. GitHub Guides [Электронный ресурс] – режим доступа: [https://guides.github.com/](https://guides.github.com/%20)
8. Write the Docs Guide [Электронный ресурс] – режим доступа: [https://www.writethedocs.org/guide/](https://www.writethedocs.org/guide/%20)
9. Модульное тестирование в Python [Электронный ресурс] – режим доступа: [https://realpython.com/python-testing/](https://realpython.com/python-testing/%20)
10. Best Practices for Python Development [Электронный ресурс] – режим доступа: [https://docs.python-guide.org/](https://docs.python-guide.org/%20)
11. Code Review Guidelines [Электронный ресурс] – режим доступа: [https://google.github.io/eng-practices/review/](https://google.github.io/eng-practices/review/%20)
12. Работа с файловой системой в Python [Электронный ресурс] – режим доступа: [https://docs.python.org/3/library/os.path.html](https://docs.python.org/3/library/os.path.html%20)

# ПРИЛОЖЕНИЯ



Рисунок 1 – программный модуль для обработки изображений



Рисунок 2 – результаты тестирования